Energy, Mines and Resources Canada
Énergie, Mines et Ressources Canada

CANMET
Canada Centre for Mineral and Energy Technology
Centre canadien de la technologie des minéraux et de l'énergie

CROSS PROGRAM SOFTWARE

V.Srajer, N.J.Stuart, C.Neumiller, M.Heximer, W.Veldhuis

March 1987

This document is an unedited interim report prepared for the ... (CANMET)

Ce document est un rapport provisoire non révisé et rédigé ... pour la diffusion de ... de documentation ... Ministère de ... Centre canadien de la technologie des minéraux et de l'énergie (CANMET)

MICROMEDIA

ENERGY RESEARCH PROGRAM
COAL RESEARCH LABORATORIES
DIVISION REPORT ERP/CRL 87-15 (TR)

87-05146/

CROSS PROGRAM SOFTWARE

by

V. Srajer*, N.J. Stuart**, C. Neumiller***, M. Heximer***, W. Veldhuis***,

ABSTRACT

The CROSS program presented in this report is an update of a previous edition of the program. The program plots the profiles of elevation, rolling resistance and speed restriction against the distance of each segment of a mine haulage route.

The step-by-step instruction and execution of the program is explained. A listing of the program in Fortran 77 is enclosed for further reference.

---

* Head, Operations Research, CANMET, CRL, Surface Mining Lab., Devon, Alb.
** Research Scientist, CANMET, CRL, Surface Mining Lab., Devon, Alberta.
*** Co-Op Students, University of Alberta, Edmonton, Alberta.

PROGRAMME "CROSS"

par

V. Strajer*, N.J. Stuart**, C. Neumiller***, M. Heximer***, W. Veldhuis***

RÉSUMÉ

Le programme CROSS décrit dans le présent rapport est une mise à jour d'une ancienne édition du programme. Le programme trace les profiles d'élévation, la résistance au roulement et la limitation de vitesse en fonction de la distance de chaque segment d'une voie de roulage.

Les instructions pas à pas pour l'exécution du programme sont données. Un listage du programme en Fortran 77 est inclus à des fins de référence.

*Directeur, Recherche opérationnelle, CANMET, LRC, Lab. d'exploitation en surface, Devon (Alberta).
**Chercheur, CANMET, LRC, Lab. d'exploitation en surface, Devon (Alberta).
***Étudiants inscrits à un programme de coopération, Université de l'Alberta, Edmonton (Alberta).

CONTENT

Page

ABSTRACT..... i
RESUME..... iii
INTRODUCTION..... 1
CROSS PROGRAM..... 1
User Interface..... 1
Output Plots..... 2
Revisions..... 2
Program Listing..... 3
CONCLUSION..... 10
REFERENCES..... 10

INTRODUCTION

The CROSS program developed earlier by the Surface Mining Laboratory (1) was revised to improve the shapes of the plots, introduced interruption during plotting to reviev the plots, and introduced other subroutines aimed at improving the presentation of the graphs.

The program was designed to plot the profiles of elevation, rolling resistance and speed restriction against the distance of each segment of mine haulage roads. The program profiles help the user to visualize mine haulage system during optimization studies to identify areas for possible modification.

CROSS PROGRAM

User Interface.

The CROSS program is found in the [IAN.FALL86.MINES] directory. To run the program, the succeeding procedure must be followed:

1. Type in RUN CROSS on the terminal connected to the Calcomp plotter. The plotter must be set up according to the Calcomp plotter user's guide.

2. "ENTER INPUT FILE". Type in the name of the mine file that is to be plotted. Do not include the .MAP suffix.

3. "RELOCATE ORIGIN FOR NEXT PLOT". Wait for present plot to be completed. This prompt appears when the plotter is almost finished with the current mine layout.

4. Press [MANUAL], [NEW PLOT], [ENTER], [RELOCATE PLOT].

5. Move to nev origin and press [ENTER].

6. Press [AUTO].

7. Press <CR> on the keyboard.

All of the prompts listed above appear on the terminal screen for each segment that is contained in the mine file.

The program uses the file <MINE>.MAP produced by the program MAP. This program does not produce an output file at this time.

Output plots.

A 9 x 22 cm plot is produced, having a title composed of the mine name and the segment name of the segment being plotted. It is divided into three sections with a y-axis of 3 cm each:

1. elevation (EL),
2. rolling resistance (RR), and
3. speed restriction (SR).

Each section is plotted in a different color. The labeling of the RR section occurs on the opposite side of the plot to improve readability. The x-axis is the distance along the segment.

Revisions.

The following revisions were made to the plot:

1. The shape of the graph was changed from an upright rectangle to a landscape rectangle.

2. The label size was increased.

3. The name of the mine and the segment were centered over the top axis and both are written on the same line.

4. The AXIS subroutine was changed to give a chart without a label or axis numbering on the opposite side of the chart.

5. The speed restriction graph reflects changes immediately. Previously, gradual changes over the segment in which the change occurs were plotted.

The following revisions were introduced to the program itself:

1. The input file name is prompted for during the execution of the program.

2. The speed restriction plotting was changed by the addition of an 'if/then' loop to facilitate immediate changes in speed restrictions.

3. The program is temporarily interrupted at the end of the first segment in the file, requiring the plotter to be set up for the next segment by relocating the origin. The plotter no longer moves immediately to the new location to plot the next segment. This interruption is repeated at the end of every segment in the file except for the last segment.

Program Listing.

The listing of the CROSS program in VAX FORTRAN 77 is as follows:

```
      program cross

c     LINK CROSS+GRAPHICS+GRAPHICS2+SPLINE

      real x(1000),y(1000),a(1000),b(1000),c(1000),d(1000),s(1001),ex(1000)
      real why(1000),firstx,firstel,firstrr,firstsr,factx,factel,factrr
      real factsr,minx,minel,minrr,minsr,maxx,maxel,maxrr,maxsr,data
      real distance(1000),EL(1000),RR(1000),dist,SR(1000),sx(1000)
      integer interel,interrr,intersr,indexel,indexrr,indexsr,n,k,eln
      integer rrn,srn,pen,point,pointd,count,counter
      character*75 namea,nameb,xaxis,xaxis1,xaxis2,yaxis1,yaxis2,yaxis3
      character*75 taxis,raxis,type
      character*20 fname
c     program to plot curves by fitting a cubic curve to given data
c     does not fit curves to data which x2<x1 (no overhang or circles)

c     x,y=data points
c     EL,RR,SR=data for elevation, rolling resistance and speed restriction
c     a,b,c,d=spline constants
c     ex,why=calculated points on each segment between two points
c     n=# of data lines in file
c     eln,rrn,srn=# of lines in each array that have been read from input
c     point,pointd=point at which data for EL,RR,SR,distance arrays
c           has been read
c     type=which array data read is placed into
c     data=data read to be placed in an array (EL,RR,SR)
c     k=# of segments each line between two points is divided into
c     namea,nameb=title of graph
c     xaxis,xaxis1,xaxis2,yaxis1,yaxis2,yaxis3,taxis,raxis=names of axes
c     firstx,firstel,firstrr,firstst=first value for x,EL,RR and SR axis
c     factx,factel,factrr,factsr=scaling factors for x,EL,RR and SR axis
c     minx,minel,minrr,minst=minimum values of x,EL,RR,SR
c     maxx,maxel,maxrr,maxsr=maximum values of x,EL,RR,SR
c     pen=the integer that indicates which graph is being plotted, eg rr
c     distance=distance from starting point
c     dist=distance between points, read from input file
c     indexel,indexrr,indexsr=last point in array which was read,
c         not interpolated
c     interel,interrr,intersr=point which is to be interpolated
c     count=# of characters in xaxis2

c     read in data from file and determine # of lines in file and
c     open a output file if necessary
      write(*,*) ' enter the name of the input file'
      read(5,'(a20)') fname
      call charcount(fname,lin)
c     open(unit=2,file=fname(1:lin)//'.crs',status='new')
      open(unit=1,file=fname(1:lin)//'.map',status='unknown')
c     read in filename and assign to graph title

      read(1,10),namea
10    format(12x,a)

c     read in distance units, assign to xaxis
```

```
      read(1,20),xaxis1
20    format(18x,a)

c     read in elevation units, assign to yaxis1

      read(1,30),yaxis1
30    format(19x,a)
      yaxis1='EL in '//yaxis1

c     read in rolling resistance units, assign to yaxis2

      read(1,40),yaxis2
40    format(28x,a)
      yaxis2='RR in '//yaxis2

c     read in speed restriction units, assign to yaxis3

      read(1,50),yaxis3
50    format(27x,a)
      yaxis3='SR in '//yaxis3

c     read in linename, assign to nameb

333   read(1,60),nameb
60    format(a)

c     initialize variables

      n=2
      eln=0
      rrn=0
      srn=0
      minx=0.
      minel=1000000.
      minrr=1000000.
      minsr=0.
      maxel=-1000000.
      maxrr=-1000000.
      maxsr=0.0
      distance(1)=0.
      indexel=0.
      indexrr=0.
      indexsr=0.
      pen=1
      taxis='_ '
      raxis='_ '

c     read in data title, it is not used

      read(1,60)

c     read in data

100   read(1,70,end=1000),point,type,data
70    format(1x,i3,1x,a2,1x,f10.3)

c     assign data to proper arrays and determine minimum and maximum values,
c     interpolate the values between the given values where necessary

      if(type.eq.'EL')then

c     assign to elevation array
```

```
            EL(point)=data

c       determine maximum and minimum values

            call minmax(EL(point),maxel,minel)

c       interpolate data, if needed

            interel=1
33          if((eln.ne.0).and.(point.ne.(interel+indexel)))then
                EL(indexel+interel)=(((EL(point)-EL(indexel))*interel)/
     3              (point-indexel))+EL(indexel)
                interel=interel+1
                goto 33

             endif
             indexel=point
             eln=eln+1
         elseif(type.eq.'RR')then
             RR(point)=data
             call minmax(RR(point),maxrr,minrr)
             interrr=1
 43          if((rrn.ne.0).and.(point.ne.interrr+indexrr))then
                 RR(indexrr+interrr)=(((RR(point)-RR(indexrr))*interrr)/
     3               (point-indexrr))+RR(indexrr)
                 interrr=interrr+1
                 goto 43
             endif
             indexrr=point
             rrn=rrn+1
         elseif(type.eq.'SR')then
             SR(point)=data
             call minmax(SR(point),maxsr,minsr)
             intersr=1
 53          if((srn.ne.0).and.(point.ne.intersr+indexsr))then
                 SR(indexsr+intersr)=SR(indexsr+intersr-1)
                 intersr=intersr+1
                 goto 53
             endif
             indexsr=point
             srn=srn+1
         elseif(type.eq.'FF')then

c       this is the flag that tells that all the additional data has been read

             goto 1000
         endif

         goto 100
1000     continue

c        read in distance title, assign to xaxis

         read(1,60),xaxis2
         call charcount(xaxis2,count)
         xaxis=xaxis2(1:count)//'in '//xaxis1

c        read in distance data, find distance for each point and find
c        distance maximum

200      read(1,80,end=2000),pointd,dist
80       format(1x,i3,1x,f7.2)
```

```
      if(pointd.eq.000)then
c     this is the flag that shows that all the distance data has been read
          goto 2000
      endif
      distance(n)=distance(n-1)+dist
      n=n+1
      goto 200
2000  continue
      n=n-1
      maxx=distance(n)
c     Plot x-axis and titles of graph, calculate xaxis scale.
      call myscale(x,n,minx,maxx,22.0,firstx,factx)
      x(n+1)=firstx
      x(n+2)=factx
      do 35,i=1,n
          x(i)=distance(i)
35    continue
c     Plot elevation axis and  calculate scale.
      call myscale(EL,n,minel,maxel,3.,firstel,factel)
      EL(n+1)=firstel
      EL(n+2)=factel
      call chart(pen,x,EL,x,EL,n,namea,nameb,xaxis,yaxis1,taxis,raxis)
      do 45, i=1,n+2
          y(i)=EL(i)
45    continue
c     perform cubic function and plot curve
c     calculate values to pass to the spline subroutine
63    call grade(x,y,n,s)
      do 25, j=1,n-1
c     calculate cubic spline coefficients
          call spline(x(j),x(j+1),y(j),y(j+1),s(j),s(j+1),a(j),
     3        b(j),c(j),d(j))
c     determine the number of segments that each line will be split into
          call segment(x,y,j,n,k)
c     calculate the values of points along the line segments
          ex(1)=x(j)
          why(1)=y(j)
          do 15, i=1,k
              ex(i+1)=x(j)+(i*(x(j+1)-x(j)))/(k)
              why(i+1)=(a(j))+(b(j)*ex(i))+(c(j)*(ex(i)**2))+(d(j)*
     3                (ex(i)**3))
15        continue
c     transfer scale parameters to cubic function calculations
c     so that the curves can be plotted
      ex(k)=x(j+1)
      why(k)=y(j+1)
```

```
            ex(k+2)=x(n+1)
            ex(k+3)=x(n+2)
            why(k+2)=y(n+1)
            why(k+3)=y(n+2)

c           plot curve

            call line(ex,why,k+1,1,0,4)

25          continue

            pen=pen+1

c           determine whether speed restriction plot or new line plot is performed

            if(pen.eq.3)then
               goto 73
            elseif(pen.eq.4)then
               goto 83
            endif

c           change pens and move pen to new origin
            call newpen(2)
            call plot(0.,3.,-3)

c           Plot rolling resistance axis and calculate scale.

            call myscale(RR,n,minrr,maxrr,3.,firstrr,factrr)
            RR(n+1)=firstrr
            RR(n+2)=factrr
            call chart(pen,x,RR,x,RR,n,namea,nameb,xaxis,raxis,taxis,yaxis2)
            do 55, i=1,n+2
               y(i)=RR(i)
55          continue
            goto 63

c           change pens and move pen to new origin

73          call newpen(3)
            call plot(0.,3.,-3)

c           Plot speed restriction axis, top axis and calculate scale.

            call myscale(SR,n,minsr,maxsr,3.,firstsr,factsr)
            SR(n+1)=firstsr
            SR(n+2)=factsr
            call chart(pen,x,SR,x,SR,n,namea,nameb,xaxis,yaxis3,taxis,raxis)
            j=1
            sx(1)=x(1)
            y(1)=SR(1)
            do 65, i=2,n
            j=j+1
            if(SR(i).ne.SR(i-1)) then
            sx(j)=x(i)
            y(j)=SR(i-1)
            j=j+1
            endif
               y(j)=SR(i)
            sx(j)=x(i)
65          continue
            sx(j+1)=x(n+1)
            sx(j+2)=x(n+2)
            y(j+1)=SR(n+1)
```

```
      y(j+2)=SR(n+2)
c     plot speed restriction lines

      call line(sx,y,j,1,0,4)

c     repeat procedure for more lines, moving origin to new piece of paper

83    if(pointd.eq.000)then
            call plot(0.0,0.0,999)
      write(*,*) ' Relocate origin for next plot'
      write(*,*) ' Wait for present plot to be completed. '
      write(*,*) ' Press [MANUAL], [NEW PLOT], [ENTER], [RELOCATE PLOT]'
      write(*,*) ' Move to new origin and press [ENTER], Press [AUTO].'
      write(*,*) ' Hit the RETURN KEY on the keyboard'
      read(*,'(a)') nameb
      call plots(0,0,6)
          goto 333
      endif

c     end of plot

      call plot(0.0,0.0,999)
      close(unit=1)
      stop
      end
c*************************************************************************

      subroutine minmax(x,maxx,minx)

c     subroutine to determine the maximum and minimum values of x

      real x,maxx,minx

c     x=data point
c     maxx=maximum of x
c     minx=minimum of x

      if(maxx.lt.x)then
          maxx=x
      endif

      if(minx.gt.x)then
          minx=x
      endif

c     return to main program

      return
      end
      subroutine chart(pen,x,y,top,r,n,namea,nameb,xaxis,
      3       yaxis,topaxis,raxis)

c     a  subroutine to plot a four sided chart ( or graph), based on the
c     requirements of the cross program

      real x(n+2),y(n+2),top(n+2),r(n+2)
      integer n,z,t,w,v,m,f,pen
      character*30 xaxis,yaxis,topaxis,raxis,namea,nameb

c     x,y=x,y data points
c     top,r=axis at top and right-hand side of page, with
c           corresponding data points
c     z,t=#of characters in title lines
c     n=# of data lines in file
```

```
c       w,v,m,f=# of characters in axis titles;  x,y,top,r, respectively
c       xaxis,yaxis,taxis,baxis = titles of each axis
c       namea,nameb=title of chart
c       pen=# of call to chart

c       determine values for z,t,w,v,m,f

        if(pen.eq.1)then
            call charcount(namea,z)
            call charcount(nameb,t)
            call charcount(xaxis,w)
            call charcount(yaxis,v)
            call charcount(raxis,f)
        elseif(pen.eq.3)then
            call charcount(yaxis,v)
            call charcount(raxis,f)
            call charcount(topaxis,m)
        else
            call charcount(yaxis,v)
            call charcount(raxis,f)
        endif

c       for plotting of first part of graph

        if(pen.eq.1)then

c       initialize and open the plotting routine
            call plots(0,0,6)

c       create a  new origin
        call plot(3.5,6.,-3)

c       plot the title of the graph
        csize=.3
        nameb=namea(1:z)//' : '//nameb(1:t)
        t=z+t+3
        call symbol(11.-(t*csize/2.),9.3,csize,nameb,14,0.,t)

c       plot the x-axis and its title

            call axis(0.,0.,xaxis,-w,22.,0.,x(n+1),x(n+2))

c       plot remaining axis and its title

        elseif(pen.eq.3)then
            call axis2(0.,3.,-22.,0.)
        endif

c       plot y and r axis (common to all calls to chart)

        if((pen.eq.1).or.(pen.eq.3))then
            call axis(0.,0.,yaxis,v,3.,90.,y(n+1),y(n+2))
            call axis2(22.,0.,3.,90.)
        else
            call axis2(0.,0.,-3.,90.)
            call axis(22.,0.,raxis,-f,3.,90.,r(n+1),r(n+2))
        endif
c       return to the main program

        return
        end
```

CONCLUSION

Revisions to the program CROSS are presented in this report to inform user's about the changes that were incorporated. The changes improved readability of the program and the program's graphics.

Instruction in the execution of the program and a nev program listing are enclosed for proper documentation of the software.

REFERENCES

1. C. Neumiller, N.J. Stuart, M. Heximer, W. Veldhuis : "Digitizing Programs to Generate Input Data for a Haulage Simulation Model by Using Fitting Cubic Curves", ERP/CRL Divissional Report 86-129 (TR), CANMET, CRL, Devon, Alberta.